**package** GraphTheory;

**import** java.util.LinkedList;

**import** java.util.Queue;

**import** java.util.Scanner;

**class** BFSDirectedGraph

{

**public int**[][] **graph**;

**public int vertex**;

**public int edge**;

**public int**[] **distance**;

**public int**[] **parent**;

**public boolean**[] **visited**;

Scanner **input**;

Queue<Integer> **queue**;

**public** BFSDirectedGraph(**int**[][] graph, **int** vertex, **int** edge) {

**this**.**graph** = graph;

**this**.**vertex** = vertex;

**this**.**edge** = edge;

**this**.**distance** = **new int**[vertex];

**this**.**parent** = **new int**[vertex];

**this**.**visited** = **new boolean**[vertex];

**this**.**input** = **new** Scanner(System.***in***);

**queue** = **new** LinkedList<>();

}

**public void** BFS()

{

**for** (**int** i=0;i<**vertex**;i++)

{

**parent**[i] = -1;

**distance**[i] = 0;

**visited**[i] = **false**;

}

**for** (**int** i=0;i<**vertex**;i++)

{

**for** (**int** j=0;j<**vertex**;j++)

{

**graph**[i][j] = 0;

}

}

**for** (**int** i=0;i<**edge**;i++)

{

**int** n1 = **input**.nextInt();

**int** n2 = **input**.nextInt();

**int** weight = **input**.nextInt();

**graph**[n1][n2] = weight;

}

System.***out***.print(**"After BFS traversal: "**);

**queue**.add(0);

**visited**[0] = **true**;

**while** (!**queue**.isEmpty())

{

**int** j = **queue**.remove();

System.***out***.print(j+**" "**);

**for** (**int** i=0;i<**vertex**;i++)

{

**if** (**graph**[j][i] !=0 && !**visited**[i])

{

**distance**[i] = **distance**[j] + **graph**[j][i];

**parent**[i] = j;

**queue**.add(i);

**visited**[i] = **true**;

}

}

}

System.***out***.println(**"---------"**);

**for** (**int** i=0;i<**vertex**;i++)

{

System.***out***.println(**"Distance "**+i+**": "**+**distance**[i]);

}

System.***out***.println(**"-------------"**);

**for** (**int** i=0;i<**vertex**;i++)

{

System.***out***.println(**"Parent "**+i+**": "**+**parent**[i]);

}

}

}

**public class** BreathFastSearchDirectedGraph {

**public static void** main(String[] args) {

**int**[][] graph = **new int**[100][100];

Scanner input = **new** Scanner(System.***in***);

System.***out***.println(**"Enter number of vertex: "**);

**int** vertex = input.nextInt();

System.***out***.println(**"Enter number of edges: "**);

**int** edge = input.nextInt();

BFSDirectedGraph object = **new** BFSDirectedGraph(graph,vertex,edge);

object.BFS();

}

}